Image recognition | Specific object recognition | Open API

Technology Reports

High-speed, Large-scale Image Recognition and API

We have developed an image recognition system that makes it possible to recognize items (objects) in a photograph by instantly searching for similar images in a large-scale database containing over five million images. By implementing this real-time image recognition algorithm, we have realized a new human interface which takes an image as an input to the system instead of text or voice data. In this article, we describe our high-speed image recognition algorithm developed by NTT DOCOMO, as well as an image recognition API provided by NTT DOCOMO.

Service Innovation Department
Hayato Akatsuka
Teppei Inomata
Toshiki Sakai

1. Introduction

Image recognition refers to technologies to identify objects within images.

In the history of image recognition, character recognition was developed first, and has been useful in improving work efficiencies in commercial and industrial fields. In Japan, with the introduction of the postal code system in 1968, Toshiba Corporation implemented automatic mail sorting equipment [1] that incorporates the first ever hand-written character recognition. The equipment mechanized the sorting of mail by postal code, which had been done by hand before this invention. More recently, as computing power has increased, development of image recognition algorithms has become more active because the image recognition requires processing power. Reduction in size and price of cameras has enabled ordinary consumers to experience the benefits of image recognition in their daily lives. For example, Toyota Motors Corporation is providing the Night-View system [2], as part of initiatives to achieve a traffic-accident-free society, using image processing. The Night-View system detects pedestrians and notifies the driver in real time in order to improve safety when driving at night. In the gaming industry, Microsoft developed the Kinect™*1 [3] for Xbox360®*2 in 2010, which enabled natural game play through gestures, without using a physical controller. In e-commerce, Amazon.com introduced real-time product identification from images using object recognition, and developed Amazon Firefly*3 [4] in 2014, which direct users to its online shopping site through object recognition. These are just a few examples, but they show how image processing has permeated our daily lives in the half-century since its introduction. In the future, as smartphones and wearable devices become more common, we expect the need for instant recognition of all kinds of objects in photographs will increase still further.

To this end, NTT DOCOMO is working to develop and improve our own image recognition technologies. We have

*1 **Kinect™:** A registered trademark or trademark of Microsoft Corp. in the United States and other countries.
*2 **Xbox360®:** A registered trademark of Microsoft Corp. and related companies.

already provided our Utsushite Honyaku[*4] service, which performs character recognition to provide foreign language translations of Japanese by simply holding the camera over the text. Currently, we are also developing an image recognition engine that can recognize more complex objects than text. For recognition of complex objects, images of the objects to be recognized had to be registered in the database beforehand, and the main task of the image recognition is to identify items (or objects) by comparing input images with images stored in the large-scale database in real time. One big challenge for image recognition is handling the large-scale database in real time. As the number of items registered in the database increases, the number of items which share similar image characteristics increases as well, and this causes a drop in recognition accuracy. Also, as the number of images registered in the database increases, it takes more time to look up items in the database, and this causes a drop in processing speed. NTT DOCOMO solved these issues that commonly exist in image recognition by improving our algorithms, and realized highly accurate image recognition from a large-scale database of several million images in less than one second. Our image recognition algorithm is based on specific object recognition.

This article describes the algorithms used in image recognition technology developed by NTT DOCOMO that recognizes items (objects) in photographs. These algorithms result in the accuracy and processing speed of the image recognition. This article also gives an overview of the image recognition Application Programming Interface (API), which NTT DOCOMO began offering in October 2010, through docomo Developer support [5], in order to create open innovation and to support developers.

2. Image Recognition Details

2.1 Image Recognition Algorithms

The image recognition algorithm used in the image recognition engine developed by NTT DOCOMO (hereinafter referred to as "the algorithm") mainly focus on objects which have distinctive patterns on their planar surfaces. It identifies what the items in the photograph are (e.g., if it is a book, then the book itself can be recognized, so that specific information about the book can be obtained). The image recognition process is divided roughly into the following three phases (**Figure 1**).

(1) Keypoint detection

Points that indicate characteristics of the object (keypoints) are extracted from the image entered by the user (the query image) in real time. The keypoints in images of objects stored in the database are similarly extracted beforehand. These images stored in the database hereinafter will be called "reference images."

(2) Image feature[*5] description

For each keypoint extracted from the query and reference images in (1), a vector describing the characteristics of the keypoint ("image features") is computed from information such as the distribution of brightness at and around the point. This process is done in real time for the query image, and beforehand, off-line for the reference images.

(3) Image feature comparison

The image features for the query and reference images are compared, and the reference image which has image features that are the most similar to those of the query image is selected.

Each of these phases is described below in more detail.

1) Keypoint Detection

To identify an object in a photograph by image recognition, image characteristics of the object must be extracted from the image data. With specific object recognition, a set of keypoints extracted from the image characterizes the object.

It is desirable that the same keypoints can be extracted invariantly from the image, regardless of various photographic conditions and shooting methods. Typically, scale-invariant keypoints appear at corners or at the intersections of lines. We have combined several corner detection methods to implement more reliable keypoint detection.

Keypoints are extracted from the

*3 **Amazon Firefly:** A trademark of Amazon.com in the United States and other countries.

*4 **Utsushite Honyaku:** The name of character recognition service provided by NTT DOCOMO.

*5 **Feature:** A feature consists of numerical values. Sets of features capture unique image characteristics of an object that can represent the object. In particular, our feature is computed based on the brightness distribution surrounding detected keypoints.

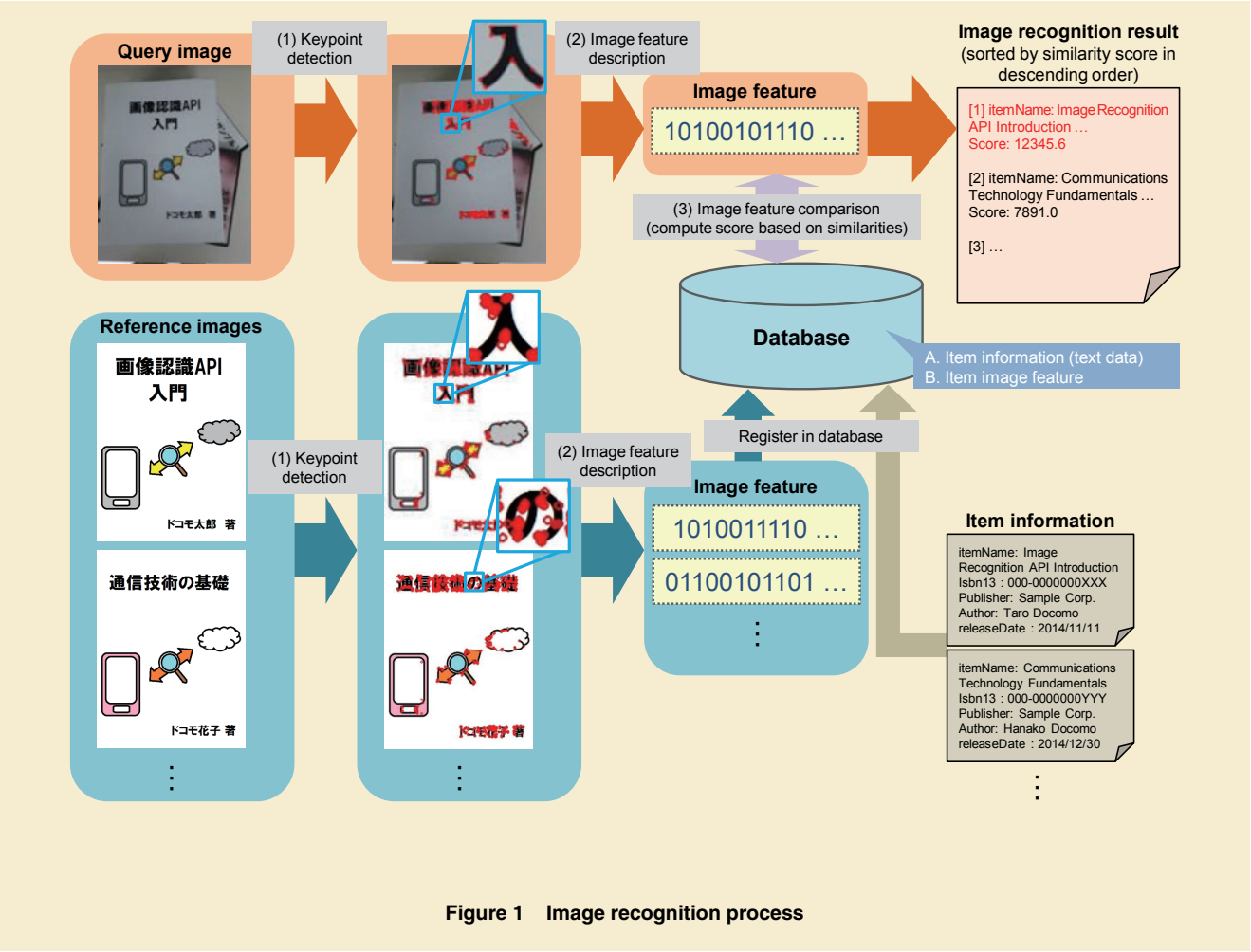

Figure 1 Image recognition process

query image in real time and from the reference images off-line beforehand. It is desirable that the same keypoints are detected in the query image and the reference images, but at certain degree of discrepancy can be expected due to differences in photographic conditions and shooting methods, even though the keypoints detection is robust.

2) Image Feature Description

An image feature is computed for each keypoint detected in the previous phase. If the objects shown in the query image and a reference image are the same, we expect many of the keypoints in the query image to correspond to keypoints in the reference image. We compute a unique image feature for each keypoint in order to compute the similarity between the query image and reference images.

The image feature used in this algorithm is typically called a "local feature" in specific object recognition. The local feature is a vector that describes the distribution of brightness in the area at and around the keypoint. Our algorithm defines the image feature such that for a given keypoint, the feature is invariant, regardless of changes in scale or rotation between photographs. Comparing image features makes it possible to find similar images.

There are several standard ways of describing local features in computer vision. One is called Scale-Invariant Feature Transform (SIFT) [6], and another is called Speeded Up Robust Features (SURF) [7]. Both algorithms produce local features that are scale and orientation invariant. For our algorithm, we have kept scale and roll invariance

as with SIFT and SURF, but coded the local feature using binary*6 vectors to speed up the image feature comparison, as described in 3).

3) Image Feature Comparison

In this phase, the set of local features from the query image are compared with those of reference images in order to retrieve highly similar objects from the database. Our algorithm is advanced compared to others in the computer vision industry today, because it can perform local feature comparisons much faster.

The database contains both product information and local features for each object. The product information includes title, author, and publication date, for example. The local features are calculated for each reference image as described in 2). These local features are used for recognition because binary comparison based on local features is much faster than brute-force comparison based on raw image data.

The local features from the query image are compared with all of the local features stored in the database in order to extract matching keypoint pairs. If the same object appears in both the query image and a reference image, many keypoint pairs will be found. After extracting matching keypoints, posture estimation is performed to eliminate a set of keypoint pairs which do not follow the majority of them. A similarity score is then computed based on the number of remaining keypoint pairs and the image feature similarity of the pairs.

With millions of reference images, computing the similarities by brute force would take more than one minute to perform just one image recognition or process just one query image.

To solve this problem, we developed a faster comparison technique using Locality Sensitive Hashing*7 (LSH). LSH summarizes local features in a hash space with fewer dimensions, so that similar data can be searched efficiently. LSH is a probabilistic search technique, so it is not theoretically guaranteed to find the optimal solution, but in most real cases, it does find an appropriate solution. It is also able to complete a comparison with several million reference images in less than one second.

2.2 Recognition Performance

In order to check the performance of our image recognition algorithm, we conducted evaluation tests. Here, we describe the details of these tests and summarize the results in terms of both recognition accuracy and processing speed.

1) Recognition Accuracy

We conducted experiments to evaluate recognition accuracy using eight different types of query images and approximately one million reference images. Types of query image include (1) *FRONT* (clear photographs of the object taken from the front, filling up the photograph frame), (2) *BLURRED* (blurred photographs), (3) *NOISY* (photographs with random noise), (4) *ROLLED* (photographs of the object taken from the front, but images are rotated along line-of-sight axis), (5) *ENLARGED* (photographs of the object taken from the front, but enlarged so that only part of object is shown in the frame), (6) *REDUCED* (photographs of the object taken from the front, but zoomed out), (7) *PANNED* (photographs of the object taken from the left), (8) *TILTED* (photographs of the object taken from below) (**Figure 2**).

After image recognition, we take the top-three items from each image recognition result, with items sorted by similarity score. If there are any correct items within in the top-three items, we increment the number of correct image recognitions by one. Then, we divide the number of correct image recognitions by the number of image recognitions attempted to calculate the recognition accuracy. Note that the number of image recognitions attempted is same as the number of query images, and the number of correct image recognitions cannot exceed the number of query images (**Figure 3**). In our evaluation results, we achieved accuracy of over 90%, and we found that some types of query image do not degrade image recognition accuracy. These included *FRONT*, *BLURRED* and *NOISY* images. Conversely, images whose appearance is different from the corresponding reference image showed degraded image recognition accuracy. These included *PANNED* and *TILTED* images. For both *ENLARGED*

*6 **Binary:** A format for expressing numerical values in base two using strings of 0s and 1s.

*7 **Hash:** A technique to map data of arbitrary size to data of fixed size. In this article, it is used to speed up the data comparisons.

and *REDUCED* images, the recognition accuracy decreases because fewer key-points are extracted from such query images. For *ENLARGED* images, even though they share similar image-feature properties with their reference images, parts of the object are not captured within the photograph. For *REDUCED* images, details in image data are lost when they are reduced and compressed (Fig. 2).

So far, we have identified two factors that contribute to decreasing recognition

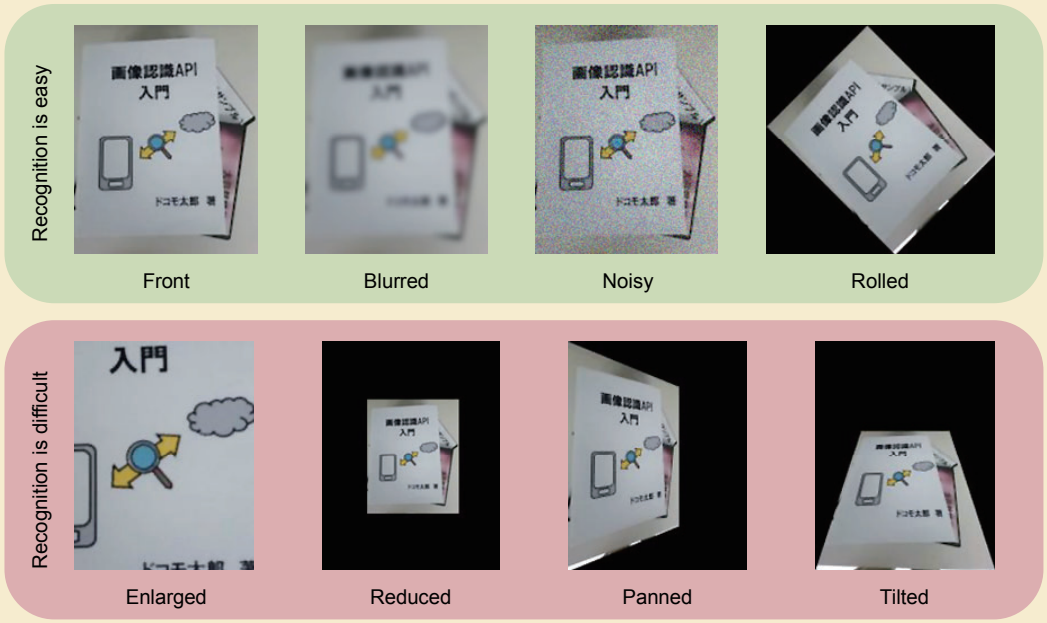

Figure 2 Eight types of query image for evaluation

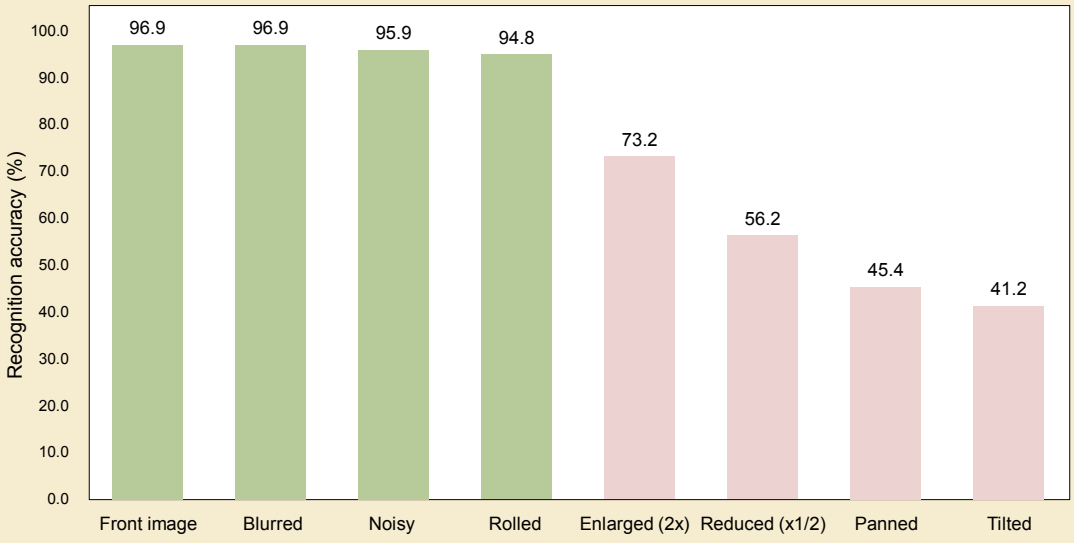

Figure 3 Recognition accuracy

accuracy in evaluation tests. The first is changes in the appearance of the object in the query images due to different scale or orientation. When the appearance of the object differs between input image and reference image, keypoints extracted from the two sources also vary. This causes fewer keypoints to be matched between the query and reference images and introduces a drop in image recognition accuracy. The second is difficulty in selecting the correct reference image when there are many similar reference images. For example, there may be a series of books whose appearance differs only in title and otherwise look the same. In such a case, since books in the series share a common appearance, image recognition accuracy can drop. NTT DOCOMO is currently working on solving these two issues in order to further improve recognition accuracy.

2) Processing Speed

For brute-force matching methods, the number of comparisons between each query image and the reference images increases in proportion to the number of reference images. However, since we have applied the high-speed matching technique using LSH to compare the query images and reference images, image features are compressed into a lower-dimensional vector space and this makes the increase in the number of comparisons much less than with brute-force matching. In the results of our evaluation tests, recognizing one query image using 100,000 reference images required an average of 0.24 seconds of processing time, while using one million reference images required an average of 0.64 seconds. Increasing the number of reference images by a factor of ten resulted in an increase in computing time by a factor of 2.7. Devices such as glasses-type wearable devices are currently spreading quickly, so we are working to further increase processing speed in order to realize more seamless recognition in real time.

3. Image Recognition API Service Details

The image recognition technology described above is provided on docomo Developer support [8] to application and service developers as "Image Recognition API". This image recognition API is a RESTful (REpresentational State Transfer) API*8 and is available to registered members of docomo Developer support.

3.1 Image Recognition API Features

"Image Recognition API" allows computers to perform image based recognition of the packaging of products sold in Japan. Books, DVDs, CDs, PC software, game software, and foods are supported. The image recognition API has a database of image and product information for more than five million products on the market. It compares the features of an input image with those of reference images in the database using the algorithm described above, and returns the product information associated with reference images similar to the query image.

Most other image recognition services, such as GAZIRU®*9 [9] provided by NEC Corporation, and the object recognition software [10] from PUX Corporation, only provide a recognition engine, and users of the image recognition service must gather the database, including images and the names of the objects in those images (information regarding what is in the images), and store it in the database on their own. Our image recognition API provides both the image recognition engine and a database of over five million items that NTT DOCOMO has gathered. Since developers can use the API with less labor for gathering data, they can develop and build mashup*10 applications and services using image recognition easily. Our API design concept is that it can be used by simply inputting an image, as described below, and developers need not be concerned with the internal image recognition processes when using it. This makes it easy to develop image recognition applications and services without any knowledge of the image recognition mechanisms.

3.2 Usage

Figure 4 shows the typical steps when using the image recognition API, from inputting an image to the image recognition API to receiving the recog-

*8 **RESTful API:** An API conforming to REST. REST is a style of software architecture developed based on design principles proposed by Roy Fielding in 2000.

*9 **GAZIRU®:** A trademark of NEC Corp.

*10 **Mashup:** To create and provide a service by combining the content and services from several other, different services.

nition result. Users of the image recognition API input a query image using the Hyper Text Transfer Protocol (HTTP) POST method*11, attaching it to the request body*12, and receive the recognition result in reply.

The result is returned as JavaScript Object Notation (JSON)*13 format text data, including the name of the product in the query image, a certainty score (similarity between query and reference image), and product details. Product details can include, for example, the publisher, publication date and author for a book, or links to e-commerce sites where the product is sold.

The API also provides end-points for feedback, so users can provide feedback on the suitability of recognition results. Feedback is used to improve recognition accuracy and to update the database.

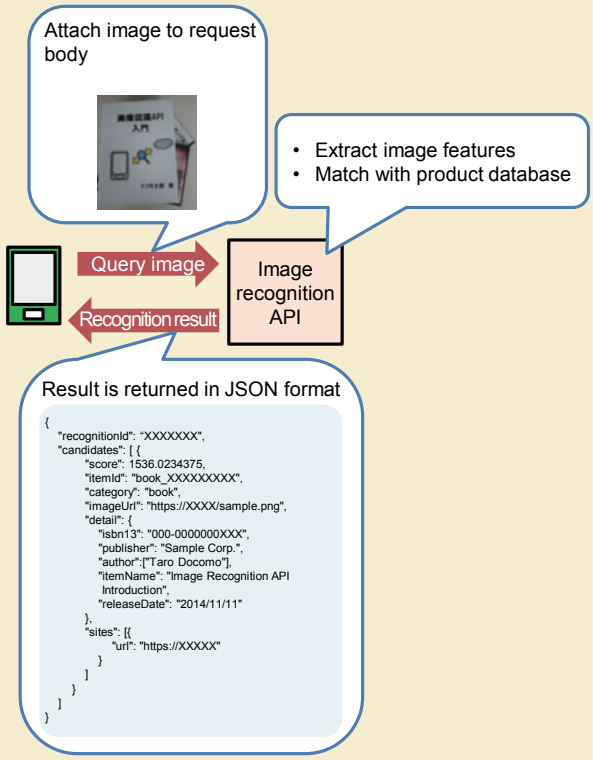

Figure 4 Image recognition process in a service (or application) with the image recognition API

3.3 Service Examples

Users of the image recognition API can develop new image recognition services by combining their own ideas with the information returned by the image recognition API. Possible examples include applications that provide product reviews and display prices retrieved from the Internet based on the product name and e-commerce site links returned from our API, or that allow users to take a picture of a product and then immediately purchase it on an e-commerce site, like Amazon Firefly [4]. Various other applications are possible, such as image based inventory management.

The image recognition API is also very compatible with Augmented Reality (AR), making it possible to recognize products and display information overlaid on the image or video based on the result of image recognition. In particular, eyeglass-type wearable devices such as Google Glass are very compatible. On these devices, image recognition could be performed on images captured with the attached camera and the information displayed on the screen of the glasses, enabling the user to get information seamlessly. After the release of the API service, developers on docomo Developer support became more active developing image recognition applications using AR technology and wearable devices.

4. Conclusion

In this article, we have described our image recognition algorithm, and our image recognition API service.

The accuracy of image recognition depends on how an object is photographed, and the speed of image recognition depends on how many objects are registered in the database. Through experiments, we have shown that high image

*11 **POST method:** A method for sending data from a client to a server when using HTTP communication.
*12 **Request body:** The part of the POST method containing the data sent from the client.
*13 **JSON:** A data description language based on JavaScript object notation.

recognition accuracy can be achieved when objects are photographed from the front, and high-speed image recognition can be achieved even on a scale of one million reference images stored in the database.

The image recognition algorithm currently developed by NTT DOCOMO mainly recognizes planar objects, but we are continuing to work on implementing high-speed, large-scale image recognition for 3D objects (such as landmarks, celebrity, clothes, and food) as well.

References

[1] Toshiba: "Toshiba Science Museum: World's First Automatic Mail Processing Equipment." http://toshiba-mirai-kagakukan.jp/learn/history/ichigoki/1967postmatter/index_j.htm

[2] Toyota Motor Corp.: "Toyota | Safety Technology | Night View." http://www.toyota.co.jp/jpn/tech/safety/technology/technology_file/active/night_view.html

[3] Microsoft Research: "Human Pose Estimation for Kinect – Microsoft Research." http://research.microsoft.com/en-us/projects/vrkinect/default.aspx

[4] Amazon.com: "Understanding Firefly - Amazon Apps & Services Developer Portal." https://developer.amazon.com/public/solutions/devices/fire-phone/docs/understanding-firefly

[5] NTT DOCOMO: "Support for Developers with docomo Developer support," NTT DOCOMO Technical Journal, Vol. 16, No. 2, pp. 48-50, Oct. 2014.

[6] D. G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

[7] H. Bay, T. Tuytelaars and L. V. Gool: "SURF: Speeded Up Robust Features," 9th European Conference on Computer Vision, 2006.

[8] NTT DOCOMO: "Image Recognition | docomo Developer support | NTT DOCOMO." https://dev.smt.docomo.ne.jp/?p=docs.api.page&api_docs_id=102

[9] NEC: "Service Implementation | GAZIRU Image Recognition Service | NEC." http://jpn.nec.com/solution/cloud/gazou/service.html

[10] PUX Corp.: "PUX Developers site." https://pds.polestars.jp/contents/technology.html